

NASA Contractor Report 195036

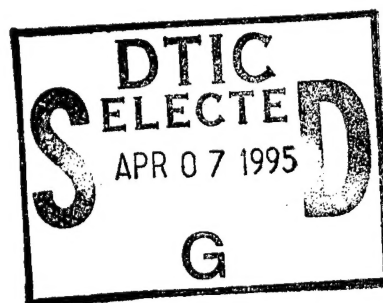
ICASE Report No. 95-8



ICASE

SEPARATION ANALYSIS, A TOOL FOR ANALYZING MULTIGRID ALGORITHMS

Sorin Costiner
Shlomo Ta'asan



Contract Nos. NAS1-18605, NAS1-19480
February 1995

19950404 106

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA 23681-0001



Operated by Universities Space Research Association

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DTIC QUALITY INSPECTION

Separation Analysis, a Tool for Analyzing Multigrid Algorithms

Sorin Costiner and Shlomo Ta'asan¹ *

Department of Applied Mathematics and Computer Science

The Weizmann Institute of Science, Rehovot 76100, Israel

and

Institute for Computer Applications in Science and Engineering

NASA Langley Research Center, Hampton, Va 23681, USA

¹ is also with: Department of Mathematics, Carnegie Mellon University, Pittsburgh, PA 15213, USA

e-mail: na.scostiner@na-net.ornl.gov

Accession For	
NTIS	CRA&I
DTIC	TAB
Unannounced	
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Abstract

The separation of vectors by multigrid (MG) algorithms is applied to the study of convergence and to the prediction of the performance of MG algorithms. The separation operator for a two level cycle algorithm is derived. It is used to analyze the efficiency of the cycle when mixing of eigenvectors occurs. In particular cases the separation analysis reduces to Fourier type analysis. The separation operator of a two level cycle for a Schrödinger eigenvalue problem, is derived and analyzed in a Fourier basis. Separation analysis gives information on how to chose relaxations and inter-level transfers. Separation analysis is a tool for analyzing and designing algorithms, and for optimizing their performance.

*This research was made possible in part by funds granted to Shlomo Ta'asan, a fellowship program sponsored by the Charles H. Revson Foundation. Both authors were supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the authors were in residence at the Institute for Computer Applications in Science and Engineering, (ICASE), Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia, 23681, USA.

1 Introduction

The efficiency of multigrid (MG) techniques in solving large scale eigenvalue problems derived from discretizations of partial differential eigenvalue problems, was shown in multiple works, for example in [1] [3] [13] [16] [18] [20] [21] [27].

This work is motivated by the need of tools of analyzing, and designing robust and efficient MG eigenvalue solvers. These tools were needed for the algorithms presented in [4], [5] [6] [19] [24] and in the reports [7] [8] [9] [10]. The algorithms were applied to electromagnetism and Schrödinger eigenvalue problems in 2D and 3D. Their efficiency resides in their eigenvector separation effectiveness, i.e., in how well they separate the subspace of desired eigenvectors from the remaining eigenvectors and how well they separate the eigenvectors inside this subspace. These naturally hint to analyze how a procedure separates given eigenvectors, i.e., how it acts on the amplitudes of the eigenvectors, or more generally on the amplitudes of a basis of vectors. The action of an algorithm on the amplitudes of the vectors of a basis is called the *separation operator*. Separation factors, showing the amplification of one vector with respect to another one, can be used to define and analyze different convergence rates. In particular cases, Fourier analysis can be used to compute separation factors. Separation analysis does not reduce to convergence analysis only; for example, one may have divergence of relaxations but good separation, and separation determines the efficiency of the algorithm. Convergence results are obtained, which can be applied directly to show the convergence of single level or multilevel algorithms. The results are used for the analysis of a two level separation operator which defines the action of a two level algorithm on the amplitudes of given vectors. For eigenvalue problems, a major issue is the “mixing” of eigenvectors due to inter-level transfers. It is shown that better convergence is obtained if the mixing is small or when all eigenvectors which get mixed are treated together, e.g., in case of clusters.

An example of separation analysis which reduces to a Fourier type analysis of the two level separation operator is presented. This is used in the design of MG eigenvector algorithms and in the prediction of their performance. For the Schrödinger eigenvalue problem with periodic boundary conditions, the analysis calculates accurately the convergence factors. The matrix of the separation operator can be computed by a subroutine and the efficiency of the algorithm can be optimized analyzing the separation operator as a function of several parameters such as relaxation types, iteration numbers.

For MG techniques and more on MG eigenvalue algorithms we refer to [2] [3] [16]. An outline of general MG approaches related to separation and MG optimization is presented in [12] [25]. For eigenvalue algorithms and theory on algebraic eigenvalue problems we refer to [17] [23] [26] [14].

2 The Separation Operator, Separation Factors and Convergence Rates

A central goal is to analyze the action of an algorithm on the amplitudes of given vectors, for example on the eigenvectors of a given matrix. This action is defined by the separation operator. The separation operator can be used in the analysis of convergence and in the optimization of the algorithm.

2.1 The Separation Operator and Separation Factors

Assume that the $n \times n$ matrix U has independent columns U_1, \dots, U_n . For example U_i may be Fourier components or the eigenvectors of a given matrix. Assume that an algorithm transforms the vector UB into the vector $U\bar{B}$, where B and \bar{B} denote vectors of dimension n . Thus B and \bar{B} are the amplitudes of U_1, \dots, U_n before and after the algorithm is applied. The algorithm defines a mapping

$$WB = \bar{B} \quad (2.1)$$

W is the mapping of the amplitudes of the vectors U_i under the action of the algorithm. The mapping W will be called the *Separation Operator*. The W may not be linear, e.g., for MG algorithms which employ projections. For particular MG cycles W is linear and will be computed in next sections.

Consider further the case when W is an $n \times n$ diagonalizable matrix with normalized eigenvectors E_1, \dots, E_n corresponding to the eigenvalues μ_1, \dots, μ_n . For simplicity assume that $|\mu_1| > |\mu_2| > \dots > |\mu_n| > 0$, although the further analysis can be performed in a similar way for the general case too. E_1 will be called the dominant eigenvector of W .

For given vectors $B = (b_1, \dots, b_n)^T$, $\bar{B} = WB = (\bar{b}_1, \dots, \bar{b}_n)^T$ define the **separation factor of U_i relative to U_j and to B** by

$$s_{ij,B} = (\bar{b}_i/\bar{b}_j)/(b_i/b_j) \quad (2.2)$$

when this is meaningful, (i.e., when $b_j \neq 0$, $\bar{b}_j \neq 0$, or define 0 and ∞ values when only one is 0). Implicitly, the $s_{ij,B}$ depends on W also.

It will be said that U_i is amplified by B if $b_i \neq 0$. Denote by e_{pm} the p 'th component of the eigenvector E_m . Define the **asymptotic separation factor of U_i relative to U_j** by:

$$s_{ij} = \mu_t/\mu_l \quad (2.3)$$

where $t = \min\{m : e_{im} \neq 0\}$ and $l = \min\{m : e_{jm} \neq 0\}$. The t and l in the definition of s_{ij} , show the first (smallest index) eigenvectors E_t and E_l which amplify U_i , respectively

U_j . It will be also said that E_t is the first eigenvector amplifying U_i . If E_t is the eigenvector amplifying first both U_i and U_j then

$$s_{ij} = 1 \quad (2.4)$$

$$s_{ij,E_t} = 1 \quad (2.5)$$

2.2 Convergence of the Iterative Algorithm

If an algorithm with separation operator W is iteratively applied starting with UB^0 , then the same results can be obtained by iterating W starting with B^0 . Thus the iteration of the algorithm reduces to a power iteration for W , just in the space of the amplitudes. The algorithm may be complicated but the power iteration is very simple (even for nonlinear W the same power iteration applies). The power iteration can be analyzed if W is known. Consider the representation of an algorithm by the iterations on W , in which the vector of amplitudes is normalized at each iteration (not necessary but to keep the norms bounded):

Fourier Power Iterations

$$B^0 = B$$

For $k = 1, 2, \dots$ **do**:

$$1) B^k = WB^{k-1}$$

$$2) B^k = B^k / \|B^k\|$$

$$3) U^k = UB^k$$

The following lemma shows towards what converge the Power Iterations and the algorithm, and shows the relative convergence rates.

Lemma 1 *In the Fourier Power Iterations algorithm, if E_1 is the dominant eigenvector of W and if B^0 is not defective in E_1 then:*

1) B^k converges to E_1 ,

2) U^k converges to UE_1 .

3) If B^0 is not defective in the first eigenvectors amplifying U_i and U_j then:

$$s_{ij,B^k} \rightarrow s_{ij} \quad (2.6)$$

Proof 1) is obvious since E_1 is the dominant eigenvector of W . Thus 2) holds. Let $B^0 = \sum_{m=1}^n a_m E_m$ with $a_1 \neq 0$. Then $B^k = \sum_{m=1}^n a_m \mu_m^k E_m$. Assume that E_t and E_l are the first eigenvectors amplifying U_i respectively U_j . The a_t and a_l are nonzero by the hypothesis 3). Then for sufficiently large k the denominator is not 0 and

$$s_{ij,B^k} = (\sum_{m=t}^n a_m \mu_m^{k+1} e_{im}) (\sum_{m=l}^n a_m \mu_m^k e_{jm}) / ((\sum_{m=l}^n a_m \mu_m^{k+1} e_{jm}) (\sum_{m=t}^n a_m \mu_m^k e_{im})) \rightarrow \mu_t / \mu_l = s_{ij}.$$

In typical situations, e.g., in eigenvalue algorithms, it is desired that U^k converges to U_1 . Different convergence rates can be defined using W , its eigenvalues and eigenvectors, and the separation factors. Define the asymptotic convergence rate of U^k towards U_1 by

$$c_1 = \sup_{B^0} \lim_{k \rightarrow \infty} \max_{j=2 \dots n} |1/s_{1j,B^k}| \quad (2.7)$$

where the B^0 in the *sup* is not defective in any of the eigenvectors.

The following corollary is obtained directly:

Corollary *In the Fourier Power Iterations U^k converges to U_1 , for any B^0 nondefective in E_1 , if and only if $E_1 = (1, 0, \dots, 0)^T$. In case of convergence, the asymptotic convergence rate is $c_1 = |\mu_2|/|\mu_1|$.*

OBSERVATIONS

The following two examples, when the convergence rate can be very good, motivate the above discussion:

1) The inverse power algorithm for eigenvalue problems:

$$AU = U\Lambda \quad (2.8)$$

where U are the eigenvectors of A associated to the eigenvalues of $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. The inverse power algorithm iterates the operator $(A - \Phi I)^{-1}$ starting with UB^0 and normalizes the result at each step, for $\Phi \approx \lambda_1$. This corresponds to a Fourier Power Type algorithm where $W = \text{diag}(1/(\lambda_i - \Phi))$, (since $(A - \Phi I)^{-1}UB = UWB = U\bar{B}$), and with another normalization at step 2) (for which a similar result can be shown). The eigenvectors of W are $E_i = U_i$ and the eigenvalues are $\mu_i = 1/(\lambda_i - \Phi)$. In this case the convergence rate is very good: $|\Phi - \lambda_1|/|\Phi - \lambda_2| \approx 0$;

2) An MG Eigenvalue Cycle, where one may expect a similarly good convergence rate since the MG cycle can be viewed as an approximation to the inverse power iteration. Such an operator will be analyzed next, for a Two Level Cycle algorithm.

3 Separation Analysis of the Two Level Cycle Algorithm

This section derives and analyses the separation operator for a two level MG cycle algebraic algorithm. The cycle has the following useful features:

1) The algorithm is algebraic and has no relation with any grid representation, neither the problems should be finer or coarser, the operators and transfers are general matrices;

- 2) The algorithm is simultaneous for several solutions, leading to a generalization of the separation setting from previous section;
- 3) The simultaneous algorithm has advantages over the corresponding sequential algorithm, e.g., it can incorporate on any level simultaneous separation techniques;
- 4) The form of the algorithm is general so that the algorithm and its analysis can be used not only for eigenvalue problems but for system solvers too.

In this cycle q solutions are treated simultaneously, thus the amplitudes matrix B has dimensions $n \times q$. The names *coarse level* and *fine level* are used only to distinguish the two levels and because these names are used often in connection with MG algorithms.

3.1 The Two Level Cycle Algorithm

Suppose that A , U , Λ are $n \times n$ matrices, and that A' , U' , Λ' are $m \times m$ matrices such that:

$$AU = U\Lambda \quad (3.9)$$

$$A'U' = U'\Lambda' \quad (3.10)$$

where U , U' are the eigenvectors of A , A' respectively, corresponding to the diagonal matrices of eigenvalues Λ and Λ' . In case of equal eigenvalues, the eigenvectors will be chosen to be linearly independent. Let P and R be two relaxation polynomials such that:

$$P(A)U = UP(\Lambda) \quad (3.11)$$

$$R(A')U' = U'R(\Lambda') \quad (3.12)$$

There are given the transfer matrices J and I such that:

$$JU = U'G \quad (3.13)$$

$$IU' = UF \quad (3.14)$$

where J and G are $m \times n$ matrices while I and F are $n \times m$ matrices.

Assume given an $n \times q$ matrix, B , a diagonal $q \times q$ matrix Φ , and an $n \times q$ matrix $U^1 = UB$ of initial solutions.

Consider the following **Two Level Cycle**:

Two Level Cycle (U^1, \bar{U})

Input $U^1 = UB$

1) Relax $U^2 = P(A)U^1$

2) Compute the residual: $S = U^2\Phi - AU^2$

- 3) Transfer the residual: $S' = JS$
- 4) Transfer the solution: $U'^1 = JU^2$
- 5) Compute the FAS right hand side: $T' = R(A')U'^1 + JS$
- 6) Solve (or Relax) the coarse level equation: $R(A')U'^2 = T'$
- 7) FAS – Correct the fine level solution: $U^3 = U^2 + I(U'^2 - JU^2)$
- 8) Relax: $\bar{U} = P(A)U^3$

Output \bar{U}

OBSERVATIONS

- 1) The step 6) can be considered as a relaxation: $U'^2 = R(A')^{-1}T'$
- 2) The algorithm can be continued on other levels in the same way.
- 3) A projection can be introduced at certain steps, multiplying the solution by a $q \times q$ matrix E . Since E depends on the solutions, the algorithm would become nonlinear in B , making the next analysis more difficult.
- 4) Different relaxations can be used at steps 1) and 8).
- 5) The FAS (Full Approximation Scheme) transfers at steps 5) and 7) follow the scheme: the problem $F'(U') = T'$ is an FAS transfer of the problem $F(U) = T$ if $T' = F'(JU) + J(T - F(U))$; and the corresponding FAS correction reads $U = U + I(U' - JU)$, see [2].

3.2 The Two Level Cycle Separation Operator

This subsection computes the separation operator W for the algebraic Two Level Cycle. The two level algorithm transforms the initial solution $U^1 = UB$ into the final solution $\bar{U} = U\bar{B}$. Since the columns of U are independent, the \bar{B} is uniquely determined for the given \bar{U} , allowing to define as in previous section

The Two Level Cycle Operator W by:

$$WB = \bar{B} \quad (3.15)$$

Denote by I_n the $n \times n$ identity matrix. Assume that $R(A')$ is invertible and denote its inverse by $R^{-1}(A')$. The operator W can be computed directly from the above algorithm and from (3.11-3.14) and is given by:

Theorem

- 1) The Two Level Cycle Separation Operator W is defined by:

$$WB = P(\Lambda)(P(\Lambda)B + FR^{-1}(\Lambda')G(P(\Lambda)B\Phi - \Lambda P(\Lambda)B)) \quad (3.16)$$

- 2) If $q = 1$ then

$$W = P(\Lambda)(I_n + FR^{-1}(\Lambda')G(\Phi I_n - \Lambda))P(\Lambda) \quad (3.17)$$

Proof Since $U^1 = UB$, the relations (3.11-3.14) imply:

$$U^2 = P(A)U^1 = P(A)UB = UP(\Lambda)B \quad (3.18)$$

$$S = U^2\Phi - AU^2 = U(P(\Lambda)B\Phi - \Lambda P(\Lambda)B) = UX \quad (3.19)$$

where

$$X = P(\Lambda)B\Phi - \Lambda P(\Lambda)B \quad (3.20)$$

The transfers give:

$$U'^1 = JU^2 = JUP(\Lambda)B = U'GP(\Lambda)B \quad (3.21)$$

$$JS = JUX = U'GX \quad (3.22)$$

and solving the coarse level equation:

$$R(A')U'^2 = R(A')U'^1 + JS \quad (3.23)$$

$$U'^2 = U'^1 + R^{-1}(A')JS = JU^2 + R^{-1}(A')JS = JU^2 + U'R^{-1}(\Lambda')GX \quad (3.24)$$

Then the FAS - Correction reads:

$$U^3 = U^2 + I(U'^2 - JU^2) = U^2 + IU'R^{-1}(\Lambda')GX \quad (3.25)$$

$$U^3 = U(P(\Lambda)B + FR^{-1}(\Lambda')GX) \quad (3.26)$$

and the final relaxation implies:

$$\bar{U} = P(A)U^3 = UP(\Lambda)(P(\Lambda)B + FR^{-1}(\Lambda')GX) = U\bar{B} \quad (3.27)$$

Hence:

$$\bar{B} = P(\Lambda)(P(\Lambda)B + FR^{-1}(\Lambda')GX) \quad (3.28)$$

Substituting X , the point 1) of the theorem is obtained:

$$WB = \bar{B} = P(\Lambda)(P(\Lambda)B + FR^{-1}(\Lambda')G(P(\Lambda)B\Phi - \Lambda P(\Lambda)B)) \quad (3.29)$$

For the case $q = 1$, Φ is a scalar and commutes with $P(\Lambda)B$ thus point 2) results.

Iterating the Two Level Cycle the following algorithm is obtained:

Two Level Iterations

Input $U^1 = UB$, Φ

for $k = 1, 2, \dots$ do:

1) Two Level Cycle (U^k, U^{k+1})

2) Normalize the columns of U^{k+1}

OBSERVATIONS

1) For the Two Level Iterations algorithm holds a Fourier Power Iterations algorithm as described in section (2).

2) As in the Corollary of Lemma 1, for $q = 1$, the convergence criteria is obtained:

Lemma 2 *In the Two Level Iterations, U^k converges to U_1 , for any initial U^1 nondefective in U_1 , if and only if W has the dominant eigenvector $(1, 0, \dots, 0)$. The single difference consists in the normalization of B^k by $B^k = B^k/c^k$ where c^k is a constant for normalizing the solutions.*

3) Denoting $B = (b_1, \dots, b_n)^T$, the k component of the vector WB is:

$$(WB)_k = P^2(\Lambda_k)b_k + P(\Lambda_k) \sum_{j=1}^n \sum_{i=1}^m (F_{ki}R^{-1}(\Lambda'_i)G_{ij})(\Phi - \Lambda_j)P(\Lambda_j)b_j \quad (3.30)$$

This formula will be used in the following sections.

3.3 Separation Factors in case of Mixing

In this paragraph the separation factors are analyzed in a case when the first two vectors of U are mixed by W during the transfers, but are not mixed with other vectors, i.e., the F and G have the structures from (3.31, 3.32). It will be shown that the mixing damages the separation, nevertheless the algorithm may be efficient for proper choices of relaxations. Assume that $q = 1$, $B = (b_1, b_2, 0, \dots, 0)^T$ and:

$$G = \begin{pmatrix} 1 & G_{12} & 0 \dots 0 \\ G_{21} & 1 & 0 \dots 0 \\ 0 & 0 & \\ \dots & & * \\ 0 & 0 & \end{pmatrix} \quad (3.31)$$

$$F = \begin{pmatrix} 1 & F_{12} & 0 \dots 0 \\ F_{21} & 1 & 0 \dots 0 \\ 0 & 0 & \\ \dots & & * \\ 0 & 0 & \end{pmatrix} \quad (3.32)$$

The first two components of WB will be:

$$(WB)_1 = P^2(\Lambda_1)b_1 + P(\Lambda_1)((R^{-1}(\Lambda'_1) + F_{12}R^{-1}(\Lambda'_2)G_{21})(\Phi - \Lambda_1)P(\Lambda_1)b_1 + (R^{-1}(\Lambda'_1)G_{12} + F_{12}R^{-1}(\Lambda'_2))(\Phi - \Lambda_2)P(\Lambda_2)b_2 \quad (3.33)$$

$$(WB)_2 = P^2(\Lambda_2)b_2 + P(\Lambda_2)((F_{21}R^{-1}(\Lambda'_1) + R^{-1}(\Lambda'_2)G_{21})(\Phi - \Lambda_1)P(\Lambda_1)b_1 + (F_{21}R^{-1}(\Lambda'_1)G_{12} + R^{-1}(\Lambda'_2))(\Phi - \Lambda_2)P(\Lambda_2)b_2 \quad (3.34)$$

Assume that the Richardson relaxation is used for A :

$$P(A) = I + \omega(A - \Phi) \quad (3.35)$$

such that

$$P(\Lambda_k) = 1 + \omega(\Lambda_k - \Phi) \approx 1 \quad (3.36)$$

and the inverse power iteration is used for A' :

$$R(A') = A' - \Phi, \quad R(\Lambda_k) = \Lambda'_k - \Phi \quad (3.37)$$

To simplify the analysis it is assumed that:

$$\Phi \approx \Lambda_1 \quad (3.38)$$

$$P(\Lambda_i) \approx 1 \quad (3.39)$$

$$(\Phi - \Lambda_2)/(\Lambda'_2 - \Phi) \approx -1 \quad (3.40)$$

$$\Lambda'_1 \approx \Lambda_1 \quad (3.41)$$

The next approximation is obtained for the first two components (the rest of the matrix is not relevant and ignored further):

$$WB \approx \begin{pmatrix} 1 & \gamma - F_{12} \\ 0 & \gamma F_{21} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (3.42)$$

Where

$$\gamma = G_{12}(\Phi - \Lambda_2)/(\Lambda'_1 - \Phi) \quad (3.43)$$

If G_{12} is not zero then γ may be large since $\Lambda'_1 - \Phi$ is small. The two eigenvalues of the above W are $\mu_1 = 1$ and $\mu_2 = \gamma F_{21}$ corresponding to the eigenvectors $E_1 = (1, 0)^T$ and $E_2 = ((-\gamma + F_{12})/(1 - \gamma F_{21}), 1)^T$. If $1 > |\mu_2|$ then iterating the two level cycle, the solutions UB^k will converge to U_1 , by Lemma 2. If $1 < |\mu_2|$ then the iterations will converge

to $(U_1, U_2)E_2$. In the case $1 < |\mu_2|$, the separation can be improved by:

- 1) improving the relaxation or the transfers (e.g., preconditioning the transfers or using higher order transfers), these leading to $1 > |\mu_2|$;
- 2) treating simultaneously the vectors U_1 and U_2 and separating them by a Rayleigh-Ritz type projection as shown for example in [5] [8] and analyzed in [10].

The asymptotic convergence rate in the case $1 > |\mu_2|$ is

$$|\mu_2/\mu_1| = |\mu_2| < 1 \quad (3.44)$$

which can be good, e.g., if $G_{12}F_{21}$ is small, but can be close to 1 if γ is large. Nevertheless the convergence in the first few cycles may be very good as shown by the relative separation factor. If at the beginning $b_1 = b_2 = 1$ and γ is large, then $s_{12,B} = (1 + \gamma - F_{12})/(\gamma F_{21}) \approx 1/F_{21}$. In this case the separation factor of U_1 relative to U_2 is approximately $1/F_{21}$ which may be very good. A better separation is obtained if $G_{12} \approx 0$, when $\gamma \approx 0$ thus the separation factor is $(1 - F_{12})/(\gamma F_{12}) \approx \infty$. In the latter case, when low mixing appears due to G_{12} , the cycle separates well the first eigenvector from the second one in the first iteration. An important observation is that in this case the role of the relaxation by P is negligible, the good separation being due mainly to the good transfer $G_{12} \approx 0$ and to the good approximation of the eigenvalue $\Lambda_1 \approx \Phi$.

3.4 Separation Factors for Decoupled Components

To show the good efficiency of the MG cycle in the case when the eigenvectors are not mixed during the transfers, the previous example in which the mixing coefficients are 0 is considered:

$$G_{12} = G_{21} = F_{12} = F_{21} = 0 \quad (3.45)$$

A more accurate estimate of the relative separation factor for $b_1 = b_2 = 1$ and by (3.33,3.34) is:

$$[1 + (\Phi - \Lambda_1)/(\Lambda'_1 - \Phi)]/[1 + (\Phi - \Lambda_2)/(\Lambda'_2 - \Phi)] = \frac{(\Lambda'_1 - \Lambda_1)(\Lambda'_2 - \Phi)}{(\Lambda'_1 - \Phi)(\Lambda'_2 - \Lambda_2)} \quad (3.46)$$

which is large in the following assumptions which are met especially in MG algorithms where Φ is obtained from coarser levels by an FMG algorithm [2] [5] [6]:

$$\Lambda'_2 - \Lambda_2 \approx 0 \quad (3.47)$$

$$(\Lambda'_1 - \Lambda_1)/(\Lambda'_1 - \Phi) \approx 1 \quad (3.48)$$

$$|\Lambda'_2 - \Phi| \gg 0 \quad (3.49)$$

OBSERVATION

The efficiency of the two level cycle may be very good also in case of close eigenvalues, when Φ approximates well the eigenvalue Λ_1 and when the two levels have close eigenvalues $\Lambda'_2 \approx \Lambda_2$. This is an explanation of the highly accurate separation obtained in numerical tests for problems presenting very close eigenvalues [4] [5] [6] [8]. The good separation obtained by the MG cycle may also explain why projections were not required on fine levels even for clusters of very close eigenvalues [4] [5] [6] [8].

4 Example of Fourier Analysis of the Two Level Separation Operator

In some cases, especially for discretizations of partial differential equations on regular grids, the two level separation operator can be computed and analyzed using for U Fourier components, as shown in the next example. This provides insight for the design of the algorithms and to the prediction of the algorithms performance.

4.1 The Two Level Separation Operator

The two level separation operator W for the algebraic two level cycle, in the common assumption $n = 2m$, is presented next. For $n = 2m$ the matrices F and G have the forms $F = (F_1, F_2)^T$, $G = (G_1, G_2)$, with $m \times m$ matrices F_1 , F_2 , G_1 , G_2 . Further, Id will denote the identity matrix. It is assumed $q = 1$, so W has the form (3.17). Denote

$$\Lambda = \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} \quad (4.50)$$

where Λ_1 , Λ_2 are $m \times m$ diagonal matrices. In this case (3.17) provides:

$$W = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \quad (4.51)$$

with

$$W_{11} = P(\Lambda_1)(Id + F_1 R^{-1}(\Lambda') G_1 (\Phi Id - \Lambda_1)) P(\Lambda_1) \quad (4.52)$$

$$W_{22} = P(\Lambda_2)(Id + F_2 R^{-1}(\Lambda') G_2 (\Phi Id - \Lambda_2)) P(\Lambda_2) \quad (4.53)$$

$$W_{12} = P(\Lambda_1) F_1 R^{-1}(\Lambda') G_2 (\Phi Id - \Lambda_2) P(\Lambda_2) \quad (4.54)$$

$$W_{21} = P(\Lambda_2) F_2 R^{-1}(\Lambda') G_1 (\Phi Id - \Lambda_1) P(\Lambda_1) \quad (4.55)$$

4.2 The Two Level Separation Operator for the Schrödinger Eigenvalue Problem

The two level separation operator for the Schrödinger Eigenvalue Problem in 1-D, with periodic boundary conditions, is derived further. For 2-D and 3-D similar derivations hold. The problem is:

$$\Delta U = U\Lambda \quad (4.56)$$

on the interval $[0, 2\pi]$. The operators A and A' are the discretizations of the Laplacian, with the stencil $(1 - 2 \ 1)/h^2$, on two grids with $2m$ (respective m) equally spaced points. The complex eigenvectors and the corresponding eigenvalues are considered:

$$U_k(l) = \exp(i2\pi kl/2m), \quad \lambda_k = -\frac{2}{h^2}(1 - \cos(2\pi k/2m)), \quad k = 0, \dots, 2m-1 \quad (4.57)$$

$$U'_k(j) = \exp(i2\pi kj/m), \quad \lambda'_k = -\frac{2}{(2h)^2}(1 - \cos(2\pi k/m)), \quad k = 0, \dots, m-1 \quad (4.58)$$

The eigenvalues are negative, ranging from $\lambda_0 = 0$ to the size of $\lambda_m = -4/h^2$, where $h = 2\pi/(2m)$ is the fine mesh size. The eigenvalues satisfy

$$\lambda_{m-k} = \lambda_{m+k}, \quad \lambda_k = \lambda_{2m-k}, \quad k = 0, \dots, m-1 \quad (4.59)$$

The real and imaginary parts of the complex eigenvectors are the real eigenvectors:

$$U_k^r(l) = \cos(i2\pi kl/2m), \quad k = 0, \dots, 2m-1 \quad (4.60)$$

$$U_k^i(l) = \sin(i2\pi kl/2m), \quad k = 1, \dots, 2m-1, \quad k \neq m \quad (4.61)$$

where $U_{2m-k}^r = U_m^r$ and $U_{2m-k}^i = -U_m^i$. The analysis can be done for the real eigenvectors but it is more convenient computationally to use the complex ones. The real eigenvectors can be used in programs which build W and analyse the separation.

The transfer J is the *full weighting* operator with stencil $(1 \ 2 \ 1)/4$. I is the linear interpolation operator. Then the matrices F_1 , F_2 , G_1 , G_2 are diagonal and all elements of F and G are zero except for $k = 0, \dots, m-1$:

$$G_{kk} = (1 + \cos(\pi k/m))/2 \quad (4.62)$$

$$G_{k,k+m} = (1 - \cos(\pi k/m))/2 \quad (4.63)$$

$$F_{kk} = (1 + \cos(\pi k/m))/2 \quad (4.64)$$

$$F_{k+m,k} = (1 - \cos(\pi k/m))/2 \quad (4.65)$$

since the transfers imply for $k = 0, \dots, m-1$, $j = 0, \dots, m-1$:

$$(U_k(2j-1) + 2U_k(2j) + U_k(2j+1))/4 = U'_k(j)(1 + \cos(\pi k/m))/2 \quad (4.66)$$

$$(U_{k+m}(2j-1) + 2U_{k+m}(2j) + U_{k+m}(2j+1))/4 = U'_k(j)(1 - \cos(\pi k/m))/2 \quad (4.67)$$

$$U_k(2j) = U'_k(j) \quad (4.68)$$

$$U_k(2j+1) = (U'_k(j) + U'_k(j+1))/2 \quad (4.69)$$

thus

$$JU_k = U'_k G_{kk}, \quad k = 0, \dots, m-1 \quad (4.70)$$

$$JU_{k+m} = U'_k G_{k,k+m}, \quad k = 0, \dots, m-1 \quad (4.71)$$

$$IU'_k = F_{kk}U_k + F_{k+m,k}U_{k+m}, \quad k = 0, \dots, m-1 \quad (4.72)$$

Then the elements of the operator W become:

$$W_{11} = \text{diag}(P(\lambda_k)(1 + F_{kk}R^{-1}(\lambda'_k)G_{kk}(\Phi - \lambda_k))P(\lambda_k)) \quad (4.73)$$

$$W_{22} = \text{diag}(P(\lambda_{k+m})(1 + F_{k+m,k}R^{-1}(\lambda'_k)G_{k,k+m}(\Phi - \lambda_{k+m}))P(\lambda_{k+m})) \quad (4.74)$$

$$W_{12} = \text{diag}(P(\lambda_k)F_{kk}R^{-1}(\lambda'_k)G_{k,k+m}(\Phi - \lambda_{k+m})P(\lambda_{k+m})) \quad (4.75)$$

$$W_{21} = \text{diag}(P(\lambda_{k+m})F_{k+m,k}R^{-1}(\lambda'_k)G_{k,k}(\Phi - \lambda_k)P(\lambda_k)) \quad (4.76)$$

4.3 Analysis and Optimization of the Two Level Cycle

The W operator (4.73-4.76) is used to analyse and optimize the two level cycle algorithm.

R is taken the shifted inverse power operator

$$R^{-1}(\lambda'_k) = 1/(\lambda'_k - \Phi) \quad (4.77)$$

The relaxation operator is taken

$$P(\Delta) = Id + \omega(\Delta - \Phi) \quad (4.78)$$

then

$$P(\lambda_k) = 1 + \omega(\lambda_k - \Phi) \quad (4.79)$$

Assume that

$$\omega = \alpha/|\lambda_{max}| = \alpha h^2/4 \quad (4.80)$$

where α will be chosen in a convenient way. To analyze the relaxation note that the asymptotic separation factor of U_i relative to U_j and to the relaxation is

$$s_{ij} = \frac{|1 + \omega(\lambda_i - \lambda)|}{|1 + \omega(\lambda_j - \lambda)|} = \frac{||\lambda_{max}| + \alpha(\lambda_i - \lambda)|}{||\lambda_{max}| + \alpha(\lambda_j - \lambda)|} = \quad (4.81)$$

$$= \left| 1 - \frac{\alpha(\lambda_j - \lambda_i)}{|\lambda_{max}| + \alpha(\lambda_j - \lambda)} \right| \quad (4.82)$$

It follows that if $\lambda_i \approx \lambda_j$ then $s_{ij} \approx 1$ thus the relaxation will be very slow in separating U_i from U_j . If λ_i is close to λ_1 and λ_j is close to λ_{max} then $s_{ij} \approx |1 - \alpha/(1 + \alpha)|$. Thus the relaxation is more efficient in separating far clusters. Within clusters another separation technique, like a Generalized Rayleigh Ritz projection, or MG projections, or an MG cycle is required, see for example [5] [8] [10]. An essential property of the relaxation is the damping of errors coming from eigenvectors which are not well represented on coarser levels. This is called the *smoothing property* of the relaxation. If U_i is a smooth vector and U_j is a component of the error, then U_j is damped by each relaxation with the separation factor s_{ij} .

Next computation finds an α such that the separation factors $s_{0k} > \beta > 1$ for all nonsmooth components U_k , assumed $\lambda = \lambda_0 = 0$. Recall from (4.57) that the nonsmooth components U_k have the frequencies $\pi/2 \leq \theta = \pi k/m \leq 3\pi/2$ corresponding to $m/2 \leq k \leq 3m/2$ and to $|\lambda_{max}| = |\lambda_m| = 4/h^2$. Then

$$|s_{0k}| = |2/(2 - \alpha(1 - \cos(\pi k/m)))| \quad (4.83)$$

The extreme values of $|s_{0k}|$ are obtained for $2\pi k/2m = \theta = \pi/2$ and π , thus $k = m/2$ respective m , for which $|s_{0m/2}| = |2/(2 - \alpha)|$ and $|s_{0m}| = |1/(1 - \alpha)|$. Both $\pi/2$ and π components can be separated by relaxation from U_0 with the factor $|s_{0m/2}| = |s_{0m}| = 3$ if

$$\alpha = 4/3 \quad (4.84)$$

Moreover, one relaxation will damp all high frequency components by a factor larger than $1/3$. If a multilevel cycle damps well the smooth components of the errors, on coarse levels where these components can be well represented, and does not amplify the oscillatory components, then the cycle should have a separation factor of $(1/3)^\mu$, for μ relaxations performed on the fine level. This factor may not be obtained in case of mixing of eigenvectors, and in the case when not all smooth components are damped well, e.g., the frequencies close to the frequencies of the desired eigenvectors, are damped slowly by the relaxation.

OBSERVATIONS

1) The WB is

$$WB = W \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} = \begin{pmatrix} W_{11}B_1 & W_{12}B_2 \\ W_{21}B_1 & W_{22}B_2 \end{pmatrix} = \begin{pmatrix} \bar{B}_1 \\ \bar{B}_2 \end{pmatrix} = \bar{B} \quad (4.85)$$

Thus if W_{21} is not small enough then the large coefficients of the smooth components, B_1 , of order 1, will imply large coefficients \bar{B}_2 of the nonsmooth errors. The factors of W_{21} which can be improved in (4.76) are

$$P(\lambda_{k+m})F_{k+m,k} \quad (4.86)$$

This can be done by changing the value of α , say close to 2, to damp more efficiently the frequencies close to $\pi/2, 3\pi/2$, which are multiplied with $1/2$ by transfers.

- 2) The relaxation can be improved also using a combination of relaxations with different values of α which will damp different frequencies.
- 3) Changing the relaxation with one having better smoothing properties, of GS type, showed large improvements in computer experiments.
- 4) Preconditioning of the transfers can be used to reduce the amplitudes of the oscillatory components. For example smoothing can be introduced before the transfer, say using a stencil $(1 \ 2 \ 1)/4$ which multiplies the Fourier components with the factor $(1 + \cos(\pi k/m))/2$.
- 5) Higher order interpolations can be used to reduce the $G_{k,k+m}$ terms.
- 6) The algorithm can be optimized using a subroutine which computes the matrix W , for different parameters such as λ , α , number of relaxation, coarse level relaxation type. An optimization search for parameter combinations providing an efficient multilevel cycle can be easily performed having such a subroutine. An observation which reduces much the analysis of W is that W has the desired eigenvector $(1, 0, \dots, 0)^T$ iff its first column is a multiple of this vector. An optimization approach can be directed towards the treatment of the first column.
- 7) An analysis similar with the one performed for the first vector can be performed for the first cluster. The elements below the block corresponding to this cluster should be 0 to avoid mixing with other clusters. The block corresponding to the cluster may have nonzero subdiagonal elements. This suggests that a separation inside the cluster is required. This separation can be performed on coarse or on fine levels.
- 8) The analysis of W can show what clusters have to be completed and which components have to be treated simultaneously. This is important for robustness and efficiency. It is simple to observe that the algorithm can be very efficient in converging to a complete cluster, treated simultaneously, while it will fail if it will treat only one of the components. A comparison between a simultaneous and a sequential algorithm can be performed using W .
- 9) If a cluster mixes with a second cluster it is an indication to include the second cluster in computation and to treat it simultaneously with the first one.
- 10) A complete cluster, which does not mix with another cluster is a good basis for a stable subspace technique.
- 11) Divergence of some components does not imply that the algorithm is not efficient. Separation is important. For example, for finding the second cluster one may have to use in computations the first cluster too. The first cluster may diverge but if the two clusters get well separated from the remaining components, then an algorithm treating both clusters simultaneously should be efficient.

- 12) Divergence can be treated using different relaxations, e.g., which amplify most specified components.
- 13) The computation of W and the optimization of the algorithm using W can be efficiently performed on coarse levels. The computation of W on fine levels is usually a very expensive task but generally not needed, a coarse version of W being sufficient for optimization.
- 14) The relative separation factors may be important in an FMG algorithm, and not as much the asymptotic factors. In an FMG algorithm for computing U_1 , the amplitudes B come already close to $(1, 0, \dots, 0)$ so that only the relative separation of certain components may be relevant, usually the ones which mix with U_1 .
- 15) The separation analysis can be useful also for cases when W is not linear. Two such cases are when a projection is used in the algorithm, and when W is a composition of separation of operators some of which can be analysed.
- 16) The computation of W can be performed in different ways. The presented way, i.e., using its analytic form, may not be the best one. A way to compute easily W for different changes of the algorithm is required. One way is to compute directly W using the action of the algorithm on a basis, e.g., on the columns of U or on the columns of the identity matrix. Another way is to compute W as a composition of simpler mappings.
- 17) The optimization can be performed by an MG procedure. This is an optimization problem with several local minima usually. See [11] for an outline and application of an MG optimization approach usable for several local minima.
- 18) The optimization of the algorithm using W leads to robust algorithms. The asymptotic convergence rates can be accurately predicted. Worst cases can be found.

The behavior predicted by W was accurately reproduced by a program implementing the two level algorithm.

References

- [1] R. E. Bank; *Analysis of a multi-level inverse iteration procedure for eigenvalue problems*; SIAM J. Numer. Anal., 19, pp. 886-898, 1982
- [2] A. Brandt; *Multigrid Techniques 1984 Guide with Applications to Fluid Dynamics*; GDM Studien Nr. 85. Gesellschaft für Mathematik und Datenverarbeitung; St. Augustin, 1984
- [3] A. Brandt, S. McCormick and J. Ruge; *Multigrid Methods for Differential Eigenproblems*; SIAM. J. SCI. STAT. COMPUT., Vol 4, No 2, 1983.

- [4] S. Costiner, F. Manolache, S. Ta'asan; *Multigrid Methods Applied to the Design of Resonant Cavities*; In print at IEEE - MTT, 1995
- [5] S. Costiner, S. Ta'asan; *Adaptive Multigrid Techniques for Large Scale Eigenvalue Problems, Solutions of the Schrödinger Problem in 2D and 3D*; submitted to Physical Review E, 1994
- [6] S. Costiner, S. Ta'asan; *Simultaneous Multigrid Techniques for Nonlinear Eigenvalue Problems, Solutions of the Nonlinear Schrödinger Poisson Eigenvalue Problem in 2D and 3D*; submitted to Physical Review E, 1994
- [7] S. Costiner, S. Ta'asan; *Separation Techniques for Multilevel Eigenvalue Solvers Applied to a Nonlinear Schrödinger Eigenvalue Problem*; Technical Report CS92-07, The Weizmann Institute of Science, Rehovot, Israel, 1992
- [8] S. Costiner, S. Ta'asan; *Multigrid Techniques for Nonlinear Eigenvalue Problems*; NASA CR-194999, ICASE Report 94-91; ICASE, NASA Langley Research Center, Hampton, VA 23681, USA, 1994
- [9] S. Costiner, S. Ta'asan; *A Robust Multilevel Simultaneous Eigenvalue Solver*; NASA CR-191491, ICASE Report 93-35; ICASE, NASA Langley Research Center, Hampton, VA 23681, USA, 1993
- [10] S. Costiner, S. Ta'asan; *The Algebraic Multigrid Projection for Eigenvalue Problems, Backrotations and Multigrid Fixed Points*; NASA CR-194990, ICASE Report 94-82; ICASE, NASA Langley Research Center, Hampton, VA 23681, U SA, 1994
- [11] S. Costiner, M. Kovalenko; *MGNN - A Multigrid Neural Network with Application to Image Processing*; Proc. of the 10th IAICVNN, Israel 1993
- [12] S. Costiner; *PhD Thesis*; The Weizmann Institute of Science, Rehovot, Israel, 1995
- [13] K. Davstad; *A Multigrid Conjugate Residual Method for the Numerical Solution of the Hartree-Fock Equation for Diatomic Molecules*; JCP, Vol 99, nr1, pp 33-38, 1992
- [14] G. H. Golub, C.F. Van Loan; *Matrix Computations*; The Johns Hopkins University Press, 1989
- [15] W. Hackbush; *On the computation of approximate eigenvalues and eigenfunctions of elliptic operators by means of a multi-grid method*; SIAM J. Numer. Anal., 16, pp 201-215, 1979

- [16] W. Hackbusch; *Multigrid methods and applications*; Springer Series in Comp. Math. 4, Springer-Verlag, Berlin, 1985
- [17] W. Kerner; *Large-Scale Complex Eigenvalue Problems, Review Article*; JCP. 85, pp 1-85, 1989
- [18] J. Mandel, S. McCormick; *A Multilevel Variational Method for $Au = \lambda Bu$ on Composite Grids*; JCP 80, pp 442-452, 1989
- [19] F. B. Manolache, S. Costiner, S. Ta'asan, D. D. Sandu; *Multigrid Subspace Techniques for the Computation of Close Frequency Eigenmodes in Complex-Shaped Resonant Cavities*; Submitted to The 13th Meeting on Hertzian Optics and Dielectrics; Zaragoza, Spain, September, 6-8, 1995
- [20] S. F. McCormick; *Multilevel Projection Methods for Partial Differential Equations*; SIAM, 1992
- [21] S. F. McCormick; *A mesh refinement method for $Ax = \lambda Bx$* ; Math. Comp., 36, pp 485-489, 1981
- [22] P.J. Nikolai; *ALGORITHM 538, Eigenvectors and Eigenvalues of Real Generalized Symmetric Matrices by Simultaneous Iteration*; ACM Trans. on Math. Software, Vol 5, No 1, pp 118-125, 1979
- [23] B. N. Parlett; *The Symmetric Eigenvalue Problem*; Prentice-Hall, Englewood Cliffs, NJ, 1980
- [24] S. Ta'asan; *Multigrid Method for Stability Calculations*; J. of Scientific Computing, Vol. 3. No. 3. 1988
- [25] S. Ta'asan, S. Costiner; *Multilevel Techniques for Large Scale Eigenvalue Problems* Proc. of the *Lanczos Centenary Conference 1993*, North Carolina State University, 1994
- [26] J.H. Wilkinson; *The Algebraic Eigenvalue Problem*; Clarendon Press, Oxford, 1965
- [27] L.Y. Zaslavsky; *An Adaptive Algebraic Multigrid for Multigroup Neutron Diffusion Reactor Core Calculations*; Applied Math. and Computation 53, pp 13-26, Elsevier, 1993

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE February 1995	3. REPORT TYPE AND DATES COVERED Contractor Report		
4. TITLE AND SUBTITLE SEPARATION ANALYSIS, A TOOL FOR ANALYZING MULTIGRID ALGORITHMS		5. FUNDING NUMBERS C NAS1-18605 C NAS1-19480 WU 505-90-52-01		
6. AUTHOR(S) Sorin Costiner Shlomo Ta'asan				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23681-0001		8. PERFORMING ORGANIZATION REPORT NUMBER ICASE Report No. 95-8		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CR-195036 ICASE Report No. 95-8		
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Dennis M. Bushnell Final Report To be submitted to the Journal of Computational Physics				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 64		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) The separation of vectors by multigrid (MG) algorithms is applied to the study of convergence and to the prediction of the performance of MG algorithms. The separation operator for a two level cycle algorithm is derived. It is used to analyze the efficiency of the cycle when mixing of eigenvectors occurs. In particular cases the separation analysis reduces to Fourier type analysis. The separation operator of a two level cycle for a Schrödinger eigenvalue problem, is derived and analyzed in a Fourier basis. Separation analysis gives information on how to chose relaxations and inter-level transfers. Separation analysis is a tool for analyzing and designing algorithms, and for optimizing their performance.				
14. SUBJECT TERMS multigrid; multilevel; eigenvalue; eigenvector; separation; Fourier analysis			15. NUMBER OF PAGES 20	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	